What Is Claimed Is:

1. A method for emulating computer viruses and/or malicious software that operates by patching additional program instructions into an emulator in order to aid in detecting a computer virus and/or malicious software within suspect code, the method comprising:

receiving the suspect code;

loading the suspect code into an emulator buffer within a data space of a computer system;

loading a first emulator extension into the emulator, the first emulator extension including program instructions that aid in the process of emulating the suspect code in order to detect a computer virus and/or malicious software;

performing an emulation using the first emulator extension and the suspect code, the emulation being performed within an insulated environment in the computer system so that the computer system is insulated from malicious actions of the suspect code; and

determining whether the suspect code is likely to exhibit malicious behavior based upon the emulation.

2. The method of claim 1, wherein loading the first emulator extension into the emulator includes loading the first emulator extension into the emulator buffer within the emulator; and

wherein performing the emulation includes emulating the program instructions that comprise the first emulator extension.

3. The method of claim 2, wherein emulating the program instructions that comprise the first emulator extension causes the emulator to

examine the suspect code looking for patterns that indicate that the suspect code is likely to exhibit malicious behavior.

4. The method of claim 2, wherein emulating the program instructions that comprise the first emulator extension causes the program instructions within the first emulator extension to facilitate emulation of the suspect code.

5. The method of claim 1, further comprising emulating the suspect code prior to loading the first emulator extension into the emulator buffer.

6. The method of claim 1, further comprising:

loading a second emulator extension into the emulator; and

performing a second emulation using the second emulator extension and the suspect code.

7. The method of claim 6, wherein the first emulator extension and the second emulator extension provide support for conflicting emulator environments.

8. The method of claim 1, wherein loading the first emulator extension involves loading the first emulator extension from a database containing a plurality of different emulator extensions.

9. The method of claim 1, wherein the first emulator extension includes code for decrypting an encrypted computer virus and other encrypted malicious code.

10. The method of claim 1, further comprising if a computer virus or other malicious software is detected within the suspect code, disinfecting the suspect code.

11. The method of claim 1, wherein the first emulator extension facilitates emulating a non-standard computer instruction opcode.

12. The method of claim 1, wherein the first emulator extension facilitates emulating an uncommonly used operating system call.

13. A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for emulating computer viruses and/or malicious software that operates by patching additional program instructions into an emulator in order to aid in detecting a computer virus and/or malicious software within suspect code, the method comprising:

receiving the suspect code;

loading the suspect code into an emulator buffer within a data space of a computer system;

loading a first emulator extension into the emulator, the first emulator extension including program instructions that aid in the process of emulating the suspect code in order to detect a computer virus and/or malicious software;

performing an emulation using the first emulator extension and the suspect code, the emulation being performed within an insulated environment in the computer system so that the computer system is insulated from malicious actions of the suspect code; and

determining whether the suspect code is likely to exhibit malicious behavior based upon the emulation.

14. The computer-readable storage medium of claim 13, wherein loading the first emulator extension into the emulator includes loading the first emulator extension into the emulator buffer within the emulator; and

wherein performing the emulation includes emulating the program instructions that comprise the first emulator extension.

15. The computer-readable storage medium of claim 14, wherein emulating the program instructions that comprise the first emulator extension causes the emulator to examine the suspect code looking for patterns that indicate that the suspect code is likely to exhibit malicious behavior.

16. The computer-readable storage medium of claim 14, wherein emulating the program instructions that comprise the first emulator extension causes the program instructions within the first emulator extension to facilitate emulation of the suspect code.

17. The computer-readable storage medium of claim 13, wherein the method further comprises emulating the suspect code prior to loading the first emulator extension into the emulator buffer.

18. The computer-readable storage medium of claim 13, wherein the method further comprises:

loading a second emulator extension into the emulator; and

performing a second emulation using the second emulator extension and the suspect code.

19. The computer-readable storage medium of claim 18, wherein the first emulator extension and the second emulator extension provide support for conflicting emulator environments.

20. The computer-readable storage medium of claim 13, wherein loading the first emulator extension involves loading the first emulator extension from a database containing a plurality of different emulator extensions.

21. The computer-readable storage medium of claim 13, wherein the first emulator extension includes code for decrypting an encrypted computer virus and other encrypted malicious code.

22. The computer-readable storage medium of claim 13, wherein if a computer virus or other malicious software is detected within the suspect code, the method further comprises disinfecting the suspect code.

23. The computer-readable storage medium of claim 13, wherein the first emulator extension facilitates emulating a non-standard computer instruction opcode.

24. The computer-readable storage medium of claim 13, wherein the first emulator extension facilitates emulating an uncommonly used operating system call.

25. An apparatus that emulates computer viruses and/or malicious software that operates by patching additional program instructions into an emulator in order to aid in detecting a computer virus and/or malicious software within suspect code, the apparatus comprising:

a loading mechanism that is configured to load the suspect code into an emulator buffer within a data space of a computer system;

wherein the loading mechanism is additionally configured to load a first emulator extension into the emulator, the first emulator extension including program instructions that aid in the process of emulating the suspect code in order to detect a computer virus and/or malicious software;

an emulation mechanism that is configured to perform an emulation using the first emulator extension and the suspect code, the emulation being performed within an insulated environment in the computer system so that the computer system is insulated from malicious actions of the suspect code; and

a determination mechanism that is configured to determine whether the suspect code is likely to exhibit malicious behavior based upon the emulation.

26. The apparatus of claim 25, wherein the loading mechanism is configured to load the first emulator extension into the emulator buffer within the emulator; and

wherein the emulation mechanism is configured to emulate the program instructions that comprise the first emulator extension.

27. The apparatus of claim 26, wherein emulating the program instructions that comprise the first emulator extension causes the emulation mechanism to examine the suspect code looking for patterns that indicate that the suspect code is likely to exhibit malicious behavior.

28. The apparatus of claim 26, wherein emulating the program instructions that comprise the first emulator extension causes the emulation mechanism to facilitate emulation of the suspect code.

29. The apparatus of claim 25, wherein the emulator is configured to emulate the suspect code prior to loading the first emulator extension into the emulator buffer.

30. The apparatus of claim 25, wherein the loading mechanism is additionally configured to:

load a second emulator extension into the emulator; and to

perform a second emulation using the second emulator extension and the suspect code.

31. The apparatus of claim 30, wherein the first emulator extension and the second emulator extension provide support for conflicting emulator environments.

32. The apparatus of claim 25, wherein the loading mechanism is configured to load the first emulator extension from a database containing a plurality of different emulator extensions.

33. The apparatus of claim 25, wherein the first emulator extension includes code for decrypting an encrypted computer virus and other encrypted malicious code.

34. The apparatus of claim 25, further comprising a disinfecting mechanism that is configured to disinfect the suspect code if a computer virus or other malicious software is detected within the suspect code.

35. The apparatus of claim 25, wherein the first emulator extension is configured to facilitate emulating a non-standard computer instruction opcode.

36. The apparatus of claim 25, wherein the first emulator extension is configured to facilitate emulating an uncommonly used operating system call.